密码朋克随身手册

密码学基本概念　密码发展脉络　密码故事

古典密码　现代密码

Cypherpunk Handbook

项目说明手册 / 赠品 / 非卖品

密码学，本质上是一种保证信息机密性的学科。它包含两个主要分支，一个是密码编码学,一个是密码分析学。

密码编码学研究如何设计密码算法，比如加解密的类型、密钥的生成等。

密码分析学主要研究如何攻破这些密码算法,比如唯密文攻击、唯明文攻击、选择明文攻击、选择密文攻击、选择文本攻击等。只有研究所有可能的密码攻破方法，才能确定一个密码算法是否足够安全，才能在设计一套密码算法时有全面的考量。

密码编码学与密码分析学密不可分，两者互相影响，不断迭代，共同推动着现代密码学的发展。

密码发展脉络

Development Of Cryptography

公元前 480 年，波斯企图攻击希腊，希腊人通过隐写术的方式提前得知了这一消息，成功构建防线抵御攻击。

公元前 404 年，斯巴达人运用了一种军用加密装置，即将羊皮纸等缠在棱形木棍上，沿着木棍方向写出明文，再将羊皮纸取下来，打乱原本的明文结构，本质是一种置换加密。
公元前 4 世纪，埃涅阿斯在其著作《工事防御》的一章中描述了密码，是目前最早的有关密码的系统性论述。

公元前 1 世纪，凯撒发明了凯撒密码，将其频繁用于军事用途中，这实际上是一种简单替换密码，简单替换密码在多个世纪以来一度被认为无法破译，直到 9 世纪阿拉伯学者阿尔·金迪（Al-Kindi）在《关于破译信息的加密手稿》中首次讲述了频率分析法。

11 世纪，我国北宋军事著作《武经总要·字验》用特定字词描述了四十余种战场上常见的状态，这其实是一种简单替换密码。

13 世纪，英国修道士罗杰·培根在著作《密文其实并不神奇》中描述了几种密文加密的方式。这本书被认为是欧洲第一本描述密码的书籍。

16 世纪，为了抵御频率分析，维吉尼亚密码诞生，美国南北战争时期被频繁使用。19 世纪，弗里德里希·卡巴斯基提出了卡巴斯基实验；20 世纪，威廉·F·弗里德曼提出了重合指数，这两种方式均可用来破译维吉尼亚密码。

17 世纪，弗朗西斯·培根发明了培根密码。

1795 年，托马斯·杰斐逊发明了一种由 36 个字母顺序不同的园盘组成的机械密码加密装置，被称为杰斐逊圆盘。

19 世纪，荷兰语言学家、密码学家奥古斯特·柯克霍夫提出柯克霍夫原则：「除了密钥之外，即使密码系统的一切均被公开，它也仍然应当是安全的」，成为现代密码学的基本原则之一。

1917 年，一次性密码本正式提出，是被明确证明无法破译的密码。冷战时期苏联频繁使用一次一密（严格来说并不是，因为苏联人加密过程中多次使用了重复密钥，这也成为英美两国破解其密文的突破口）。

1918 年 3 月，ADFGVX 密码诞生，是一战时期德军使用的密码。同年 6 月，被法国密码学家乔治斯·佩恩文破解。

1929 年，美国数学家莱斯特·希尔发明了希尔密码，首次用数学代数结构描述了密码结构。

1961 年，杰克·莱文用已知明文攻击破解了希尔密码。

20 世纪 20 年代，密码机 Enigma 诞生，后被德军改造用于二战加密。20 世纪 30 年代到 40 年代，波兰以及英国两国的科学家致力于破解 Enigma。

同时二战时期，日本先后使用红密、橙密、紫密等密码机进行加密，但先后被英国、

美国破解。

1945 年，香农提出「混淆和扩散」的概念，成为现代分组密码的基本设计思想。

20 世纪后期，人类社会进入计算机时代。

1976 年，为了解决对称密码的密钥传送问题，迪菲 - 赫尔曼密钥交换体制诞生。

1977 年，分组密码 DES 诞生，是首个公司（IBM）与国家机构（NSA）合作，并成为美国国家标准的密码。

同年，首个公钥密码 RSA 诞生，自此密码算法被分成两类并行发展——公钥密码和私钥密码。

1983 年，威斯纳、本尼特、布拉萨德和布莱德巴特合写的《量子密码学或不可伪造的地铁代币》论文中，首次提出了「量子密码」的概念。

1985 年，公钥密码椭圆曲线密码诞生。

1998 年，DES 被计算机暴力破解，用时 56 小时。

1999 年，这一时间被进一步缩短到 22 小时 15 分钟。

2001 年，对称密码 AES 诞生，取代 DES 和 3-DES 成为新的加密标准。

2016 年，NIST 发布招募后量子公钥密码标准，旨在抵抗可能到来的量子时代对目前公钥算法的威胁。

未尽……

密码故事

Story Of Cryptography

比尔密码 Beale Ciphers

1800 年，一个名叫托马斯·J·比尔的人带领一支 30 多人的冒险队伍在今科罗拉多州的矿区挖矿石。后来团队成员委托比尔将挖到的这些宝藏埋在安全的地方，宝藏地点可能位于今天的弗吉尼亚州。比尔将宝藏的地点、描述和拥有者及其亲属的姓名分别加密，放到了一个铁盒子里。

1822 年，比尔来到弗吉尼亚州的林奇堡市，将铁盒子交给了一个叫罗伯特·莫里斯的旅店主人。几个月后，比尔告诉莫里斯，他的一个同伴会将密钥发给他。但密钥并未如期而至。

1845 年，莫里斯打开了盒子，他发现了那三封密文和比尔写的两封信。遗憾的是，莫里斯并没有破译出三封密文。后来这些东西被转移到了他的一个朋友手里，但这位朋友也仅仅破译了第二封密文。

1885 年，他将信件、密文一起出版，命名为 The Beale ciphers ，这便是著名的比尔密码。

比尔密码第一部分

“

71, 194, 38, 1701, 89, 76, 11, 83, 1629, 48, 94, 63, 132, 16, 111, 95, 84, 341, 975, 14, 40, 64, 27, 81, 139, 213, 63, 90, 1120, 8, 15, 3, 126, 2018, 40, 74, 758, 485, 604, 230, 436, 664, 582, 150, 251, 284, 308, 231, 124, 211, 486, 225, 401, 370, 11, 101, 305, 139, 189, 17, 33, 88, 208, 193, 145, 1, 94, 73, 416, 918, 263, 28, 500, 538, 356, 117, 136, 219, 27, 176, 130, 10, 460, 25, 485, 18, 436, 65, 84, 200, 283, 118, 320, 138, 36, 416, 280, 15, 71, 224, 961, 44, 16, 401, 39, 88, 61, 304, 12, 21, 24, 283, 134, 92, 63, 246, 486, 682, 7, 219, 184, 360, 780, 18, 64, 463, 474, 131, 160, 79, 73, 440, 95, 18, 64, 581, 34, 69, 128, 367, 460, 17, 81, 12, 103, 820, 62, 116, 97, 103, 862, 70, 60, 1317, 471, 540, 208, 121, 890, 346, 36, 150, 59, 568, 614, 13, 120, 63, 219, 812, 2160, 1780, 99, 35, 18, 21, 136, 872, 15, 28, 170, 88, 4, 30, 44, 112, 18, 147, 436, 195, 320, 37, 122, 113, 6, 140, 8, 120, 305, 42, 58, 461, 44, 106, 301, 13, 408, 680, 93, 86, 116, 530, 82, 568, 9, 102, 38, 416, 89, 71, 216, 728, 965, 818, 2, 38, 121, 195, 14, 326, 148, 234, 18, 55, 131, 234, 361, 824, 5, 81, 623, 48, 961, 19, 26, 33, 10, 1101, 365, 92, 88, 181, 275, 346, 201, 206, 86, 36, 219, 324, 829, 840, 64, 326, 19, 48, 122, 85, 216, 284, 919, 861, 326, 985, 233, 64, 68, 232, 431, 960, 50, 29, 81, 216, 321, 603, 14, 612, 81, 360, 36, 51, 62, 194, 78, 60, 200, 314, 676, 112, 4, 28, 18, 61, 136, 247, 819, 921, 1060, 464, 895, 10, 6, 66, 119, 38, 41, 49, 602, 423, 962, 302, 294, 875, 78, 14, 23, 111, 109, 62, 31, 501, 823, 216, 280, 34, 24, 150, 1000, 162, 286, 19, 21, 17, 340, 19, 242, 31, 86, 234, 140, 607, 115, 33, 191, 67, 104, 86, 52, 88, 16, 80, 121, 67, 95, 122, 216, 548, 96, 11, 201, 77, 364, 218, 65, 667, 890, 236, 154, 211, 10, 98, 34, 119, 56, 216, 119, 71, 218, 1164, 1496, 1817, 51, 39, 210, 36, 3, 19, 540, 232, 22, 141, 617, 84, 290, 80, 46, 207, 411, 150, 29, 38, 46, 172, 85, 194, 39, 261, 543, 897, 624, 18, 212, 416, 127, 931, 19, 4, 63, 96, 12, 101, 418, 16, 140, 230, 460, 538, 19, 27, 88, 612, 1431, 90, 716, 275, 74, 83, 11, 426, 89, 72, 84, 1300, 1706, 814, 221, 132, 40, 102, 34, 868, 975, 1101, 84, 16, 79, 23, 16, 81, 122, 324, 403, 912, 227, 936, 447, 55, 86, 34, 43, 212, 107, 96, 314, 264, 1065, 323, 428, 601, 203, 124, 95, 216, 814, 2906, 654, 820, 2, 301, 112, 176, 213, 71, 87, 96, 202, 35, 10, 2, 41, 17, 84, 221, 736, 820, 214, 11, 60, 760.

”

比尔密码第二部分

“

115, 73, 24, 807, 37, 52, 49, 17, 31, 62, 647, 22, 7, 15, 140, 47, 29, 107, 79, 84, 56, 239, 10, 26, 811, 5, 196, 308, 85, 52, 160, 136, 59, 211, 36, 9, 46, 316, 554, 122, 106, 95, 53, 58, 2, 42, 7, 35, 122, 53, 31, 82, 77, 250, 196, 56, 96, 118, 71, 140, 287, 28, 353, 37, 1005, 65, 147, 807, 24, 3, 8, 12, 47, 43, 59, 807, 45, 316, 101, 41, 78, 154, 1005, 122, 138, 191, 16, 77, 49, 102, 57, 72, 34, 73, 85, 35, 371, 59, 196, 81, 92, 191, 106, 273, 60, 394, 620, 270, 220, 106, 388, 287, 63, 3, 191, 122, 43, 234, 400, 106, 290, 314, 47, 48, 81, 96, 26, 115, 92, 158, 191, 110, 77, 85, 197, 46, 10, 113, 140, 353, 48, 120, 106, 2, 607, 61, 420, 811, 29, 125, 14, 20, 37, 105, 28, 248, 16, 159, 7, 35, 19, 301, 125, 110, 486, 287, 98, 117, 511, 62, 51, 220, 37, 113, 140, 807, 138, 540, 8, 44, 287, 388, 117, 18, 79, 344, 34, 20, 59, 511, 548, 107, 603, 220, 7, 66, 154, 41, 20, 50, 6, 575, 122, 154, 248, 110, 61, 52, 33, 30, 5, 38, 8, 14, 84, 57, 540, 217, 115, 71, 29, 84, 63, 43, 131, 29, 138, 47, 73, 239, 540, 52, 53, 79, 118, 51, 44, 63, 196, 12, 239, 112, 3, 49, 79, 353, 105, 56, 371, 557, 211, 515, 125, 360, 133, 143, 101, 15, 284, 540, 252, 14, 205, 140, 344, 26, 811, 138, 115, 48, 73, 34, 205, 316, 607, 63, 220, 7, 52, 150, 44, 52, 16, 40, 37, 158, 807, 37, 121, 12, 95, 10, 15, 35, 12, 131, 62, 115, 102, 807, 49, 53, 135, 138, 30, 31, 62, 67, 41, 85, 63, 10, 106, 807, 138, 8, 113, 20, 32, 33, 37, 353, 287, 140, 47, 85, 50, 37, 49, 47, 64, 6, 7, 71, 33, 4, 43, 47, 63, 1, 27, 600, 208, 230, 15, 191, 246, 85, 94, 511, 2 270, 20, 39, 7, 33, 44, 22, 40, 7, 10, 3, 811, 106, 44, 486, 230, 353, 211, 200, 31, 10, 38, 140, 297, 61, 603, 320, 302, 666, 287, 2, 44, 33, 32, 511, 548, 10, 6, 250, 557, 246, 53, 37, 52, 83, 47, 320, 38, 33, 807, 7, 44, 30, 31, 250, 10, 15, 35, 106, 160, 113, 31, 102, 406, 230, 540, 320, 29, 66, 33, 101, 807, 138, 301, 316, 353, 320, 220, 37, 52, 28, 540, 320, 33, 8, 48, 107, 50, 811, 7, 2, 113, 73, 16, 125, 11, 110, 67, 102, 807, 33, 59, 81, 158, 38, 43, 581, 138, 19, 85, 400, 38, 43, 77, 14, 27, 8, 47, 138, 63, 140, 44, 35, 22, 177, 106, 250, 314, 217, 2, 10, 7, 1005, 4, 20, 25, 44, 48, 7, 26, 46, 110, 230, 807, 191, 34, 112, 147, 44, 110, 121, 125, 96, 41, 51, 50, 140, 56, 47, 152, 540, 63, 807, 28, 42, 250, 138, 582, 98, 643, 32, 107, 140, 112, 26, 85, 138, 540, 53, 20, 125, 371, 38, 36, 10, 52, 118, 136, 102, 420, 150, 112, 71, 14, 20, 7, 24, 18, 12, 807, 37, 67, 110, 62, 33, 21, 95, 220, 511, 102, 811, 30, 83, 84, 305, 620, 15, 2, 108, 220, 106, 353, 105, 106, 60, 275, 72, 8, 50, 205, 185, 112, 125, 540, 65, 106, 807, 188, 96, 110, 16, 73, 33, 807, 150, 409, 400, 50, 154, 285, 96, 106, 316, 270, 205, 101, 811, 400, 8, 44, 37, 52, 40, 241, 34, 205, 38, 16, 46, 47, 85, 24, 44, 15, 64, 73, 138, 807, 85, 78, 110, 33, 420, 505, 53, 37, 38, 22, 31, 10, 110, 106, 101, 140, 15, 38, 3, 5, 44, 7, 98, 287, 135, 150, 96, 33, 84, 125, 807, 191, 96, 511, 118, 440, 370, 643, 466, 106, 41, 107, 603, 220, 275, 30, 150, 105, 49, 53, 287, 250, 208, 134, 7, 53, 12, 47, 85, 63, 138, 110, 21, 112, 140, 485, 486, 505, 14, 73, 84, 575, 1005, 150, 200, 16, 42, 5, 4, 25, 42, 8, 16, 811, 125, 160, 32, 205, 603, 807, 81, 96, 405, 41, 600, 136, 14, 20, 28, 26, 353, 302, 246, 8, 131, 160, 140, 84, 440, 42, 16, 811, 40, 67, 101, 102, 194, 138, 205, 51, 63, 241, 540, 122, 8, 10, 63, 140, 47, 48, 140, 288.

”

比尔密码第三部分

“

317, 8, 92, 73, 112, 89, 67, 318, 28, 96, 107, 41, 631, 78, 146, 397, 118, 98, 114, 246, 348, 116, 74, 88, 12, 65, 32, 14, 81, 19, 76, 121, 216, 85, 33, 66, 15, 108, 68, 77, 43, 24, 122, 96, 117, 36, 211, 301, 15, 44, 11, 46, 89, 18, 136, 68, 317, 28, 90, 82, 304, 71, 43, 221, 198, 176, 310, 319, 81, 99, 264, 380, 56, 37, 319, 2, 44, 53, 28, 44, 75, 98, 102, 37, 85, 107, 117, 64, 88, 136, 48, 154, 99, 175, 89, 315, 326, 78, 96, 214, 218, 311, 43, 89, 51, 90, 75, 128, 96, 33, 28, 103, 84, 65, 26, 41, 246, 84, 270, 98, 116, 32, 59, 74, 66, 69, 240, 15, 8, 121, 20, 77, 89, 31, 11, 106, 81, 191, 224, 328, 18, 75, 52, 82, 117, 201, 39, 23, 217, 27, 21, 84, 35, 54, 109, 128, 49, 77, 88, 1, 81, 217, 64, 55, 83, 116, 251, 269, 311, 96, 54, 32, 120, 18, 132, 102, 219, 211, 84, 150, 219, 275, 312, 64, 10, 106, 87, 75, 47, 21, 29, 37, 81, 44, 18, 126, 115, 132, 160, 181, 203, 76, 81, 299, 314, 337, 351, 96, 11, 28, 97, 318, 238, 106, 24, 93, 3, 19, 17, 26, 60, 73, 88, 14, 126, 138, 234, 286, 297, 321, 365, 264, 19, 22, 84, 56, 107, 98, 123, 111, 214, 136, 7, 33, 45, 40, 13, 28, 46, 42, 107, 196, 227, 344, 198, 203, 247, 116, 19, 8, 212, 230, 31, 6, 328, 65, 48, 52, 59, 41, 122, 33, 117, 11, 18, 25, 71, 36, 45, 83, 76, 89, 92, 31, 65, 70, 83, 96, 27, 33, 44, 50, 61, 24, 112, 136, 149, 176, 180, 194, 143, 171, 205, 296, 87, 12, 44, 51, 89, 98, 34, 41, 208, 173, 66, 9, 35, 16, 95, 8, 113, 175, 90, 56, 203, 19, 177, 183, 206, 157, 200, 218, 260, 291, 305, 618, 951, 320, 18, 124, 78, 65, 19, 32, 124, 48, 53, 57, 84, 96, 207, 244, 66, 82, 119, 71, 11, 86, 77, 213, 54, 82, 316, 245, 303, 86, 97, 106, 212, 18, 37, 15, 81, 89, 16, 7, 81, 39, 96, 14, 43, 216, 118, 29, 55, 109, 136, 172, 213, 64, 8, 227, 304, 611, 221, 364, 819, 375, 128, 296, 1, 18, 53, 76, 10, 15, 23, 19, 71, 84, 120, 134, 66, 73, 89, 96, 230, 48, 77, 26, 101, 127, 936, 218, 439, 178, 171, 61, 226, 313, 215, 102, 18, 167, 262, 114, 218, 66, 59, 48, 27, 19, 13, 82, 48, 162, 119, 34, 127, 139, 34, 128, 129, 74, 63, 120, 11, 54, 61, 73, 92, 180, 66, 75, 101, 124, 265, 89, 96, 126, 274, 896, 917, 434, 461, 235, 890, 312, 413, 328, 381, 96, 105, 217, 66, 118, 22, 77, 64, 42, 12, 7, 55, 24, 83, 67, 97, 109, 121, 135, 181, 203, 219, 228, 256, 21, 34, 77, 319, 374, 382, 675, 684, 717, 864, 203, 4, 18, 92, 16, 63, 82, 22, 46, 55, 69, 74, 112, 134, 186, 175, 119, 213, 416, 312, 343, 264, 119, 186, 218, 343, 417, 845, 951, 124, 209, 49, 617, 856, 924, 936, 72, 19, 28, 11, 35, 42, 40, 66, 85, 94, 112, 65, 82, 115, 119, 236, 244, 186, 172, 112, 85, 6, 56, 38, 44, 85, 72, 32, 47, 73, 96, 124, 217, 314, 319, 221, 644, 817, 821, 934, 922, 416, 975, 10, 22, 18, 46, 137, 181, 101, 39, 86, 103, 116, 138, 164, 212, 218, 296, 815, 380, 412, 460, 495, 675, 820, 952.

”

看上去完全不知所云。那么第二封密文是如何被破译的?

第二封密文的每一个数字对应的都是《美国独立宣言》的第 N 个词中的首字母。比如第一个数字 115 对应的就是第 115 个词 instituted 中首字母 I 。

> When in the Course of human Events, it becomes necessary for one People to dissolve the Political Bands which have connected them with another, and to assume among the Powers of the Earth, the separate and equal Station to which the Laws of Nature and of Nature’ s God entitle them, a decent Respect to the Opinions of Mankind requires that they should declare the causes which impel them to the Separation. We hold these Truths to be self-evident, that all Men are created equal, that they are endowed by their Creator with certain unalienable Rights, that among these are Life, Liberty, and the pursuit of Happiness—That to secure these Rights, Governments are instituted among Men, deriving their just Powers from the Consent of the Governed, that whenever any Form of Government becomes destructive of these Ends, it is the Right of the People to alter or to abolish it, and to institute new Government, laying its Foundation on such Principles, and organizing its Powers in such Form, as to them shall seem most likely to effect their Safety and Happiness.

按照这样的规则，第二封密文真正的意思便浮现出水面了。

“

I have deposited in the county of Bedford, about four miles from Buford's, in an excavation or vault, six feet below the surface of the ground, the following articles, belonging jointly to the parties whose names are given in number three, herewith:
The first deposit consisted of ten hundred and fourteen pounds of gold, and thirty-eight hundred and twelve pounds of silver, deposited Nov. eighteen nineteen. The second was made Dec. eighteen twenty-one, and consisted of nineteen hundred and seven pounds of gold, and twelve hundred and eighty-eight of silver; also jewels, obtained in St. Louis in exchange to save transportation, and valued at thirteen thousand dollars.
The above is securely packed in iron pots, with iron covers. The vault is roughly lined with stone, and the vessels rest on solid stone, and are covered with others. Paper number one describes the exact locality of the vault, so that no difficulty will be had in finding it.

我在贝德福德郡，距 Buford 酒馆约 4 英里、6 英尺深的洞穴或地窖埋藏了三号密码指定的一些人所拥有的物品：1819 年 11 月第一次存入的物品有 1014 磅金和 3812 磅银。1821 年 12 月第二次存入的物品有 1907 磅金和 1288 磅银，还有总值约 13000 美元的珠宝，在圣路易斯用银换取，以便运输。这些物品储存在带有铁盖的铁罐内。洞穴的内壁衬着石头。容器放置在结实的石壁上，并被其他物件覆盖。一号密码指明了藏宝处的地点，所以很容易就能找到它。

”

既然第二封密文的密钥是《美国独立宣言》，人们猜测第一封和第三封的密钥是不是也是某一个文本呢？但这么长时间以来，试遍很多文本，都没有成功。

与此同时，针对比尔密码的真实性也引发了更多的讨论。人们从词源、写作习惯、写作风格等方式考量，均发现了一些可疑的地方。

但无论结果如何，人们对于寻宝的热情从未消退。有人亲自去可能的地方寻找比尔留下的宝藏，却一无所获。

黄道十二宫杀手 Zodiac Killer

1968 年至 1969 年，美国加利福尼亚州北部发生了四起命案。这四起命案中，先后有 7 人被杀害，其中两人生还。

犯下这些罪行的凶手是一个自称为「十二宫」的人。在警方苦苦排查犯罪嫌疑人时，「十二宫」还在不断给报社写信，声称要杀害更多的人。

50 多年已经过去，凶手至今未被抓捕归案。和他的身份一同成为谜题的，还有几封他寄给报社的信件。

这些信件均由密码写成。

第一封密码信件于 1968 年 8 月 1 日被分成三份寄给了三家报社。因为含有 408 个字符，故被称为 Z408 。

Z408 是一种简单替换密码，于同年 8 月 8 日被唐纳德和贝蒂 . 哈德夫妇破解。

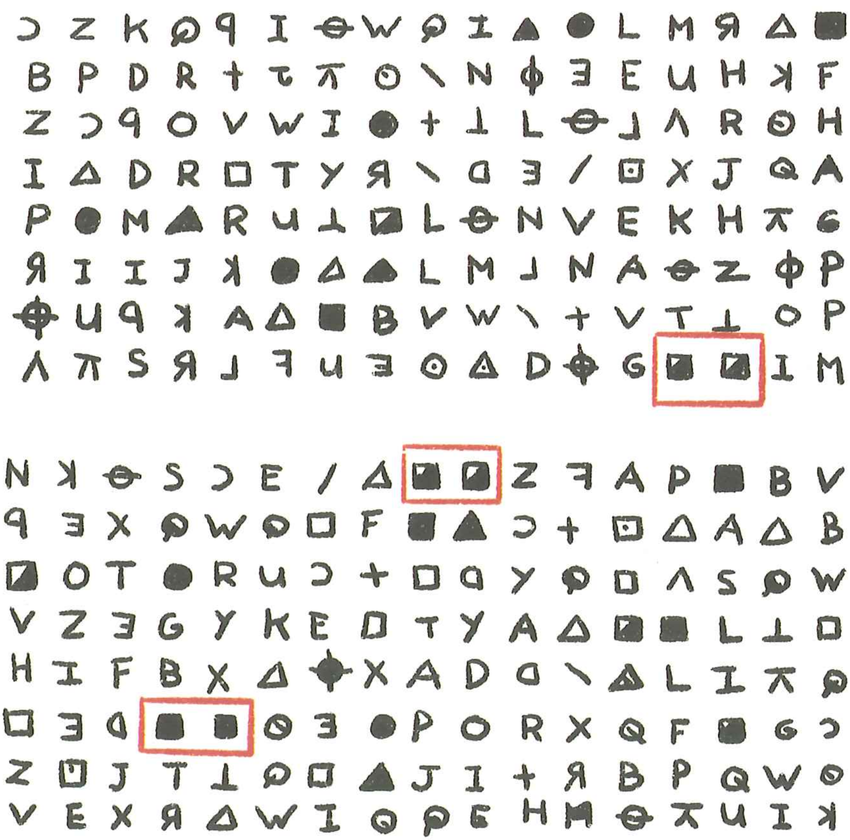
△ ◩ P / Z / U B ◩ ꓘ O R ⊼ 9 X ⊼ B
W V + Ǝ 6 Y F ⊙ △ H P ⊡ K I ⍉ Y Ǝ
M J Y ⋀ U I ꓘ ◬ ⍉ T ⊥ N Q Y D ● ⊖
S Φ / ◬ ■ B P O R A U ◩ ꟻ R ⅃ ⍉ E
ꓘ ⋀ L M Z J ꓷ Я \ 9 F H V W Ǝ ▲ Y
⊡ + ⍉ G D ◬ K I ⊖ ⊙ ⍉ X ▲ ● ⌖ S Φ
R N ⊥ I Y E ⅃ O ▲ ⍉ G B T Q S ■ B
L ꓷ / P ■ B ⊡ X ⍉ E H M U ⋀ R R ꓘ

Ɔ Z K ⍉ 9 I ⊖ W ⍉ I ▲ ● L M Я △ ■
B P D R + τ ⊼ ⊙ \ N Φ Ǝ E U H ꓘ F
Z Ɔ 9 O V W I ● + ⊥ L ⊖ ⅃ ⋀ R ⊙ H
I ◬ D R □ T Y Я \ ꓷ Ǝ / ⊡ X J Q A
P ● M ▲ R U ⊥ ◩ L ⊖ N V E K H ⊼ 6
Я I I J ꓘ ● ◬ ▲ L M ⅃ N A ⊖ Z Φ P
⌖ U 9 ꓘ A △ ■ B V W \ + V T ⊥ O P
⋀ ⊼ S Я ⅃ ꟻ U Ǝ ⊙ ◬ D ⌖ G ◩ ◩ I M

N ꓘ ⊖ S Ɔ E / △ ◩ ◩ Z ꟻ A P ■ B V
9 Ǝ X ⍉ W ⍉ □ F ■ ▲ Ɔ + ⊡ △ A △ B
◩ O T ● R U Ɔ + □ ꓷ Y ⍉ □ ⋀ S ⍉ W
V Z Ǝ G Y K E □ T Y A △ ◩ ■ L ⊥ □
H I F B X △ ⌖ X A D ꓷ \ ◬ L I ⊼ ⍉
□ Ǝ ꓷ ■ ■ ⊘ Ǝ ● P O R X Q F ■ G Ɔ
Z ⊡ J T ⊥ ⍉ □ ▲ J I + Я B P Q W ⊙
V E X Я △ W I ⊘ ⍉ E H M ⊖ ⊼ U I ꓘ

Z408 有 54 种符号，而英文字母只有 26 个。这也意味着同一个字母对应着 N 个符号，这样就打破了频率分析。

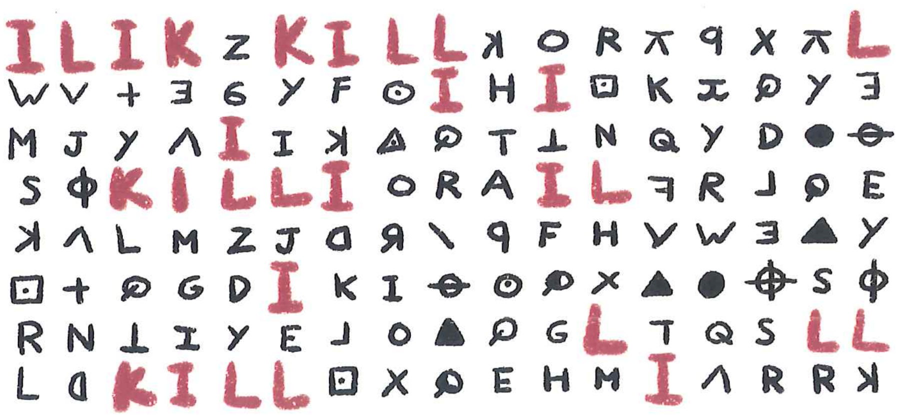
但夫妇俩发现密文中存在很多对连续出现的符号，如图所示：



在英语中，双字母出现频率最多的，是LL。

那么这两个符号有没有可能是 ll 呢？出现 ll 的，会不会是 kill 呢？

顺着这样的猜测，哈德夫妇找到了出现了 l 并且有同样符号的四个符号的组合，将其替换为 kill，这样一来，密文就变成了这样：



贝蒂大胆猜测，第一行的前几个符号说的可能是「I like killing」。这样一来，Z 变成了 E。

最终，哈登夫妇破解的译文如下：

“

I like killing people because it is so much fun it is more fun than killing wild game in the forrest because man is the most dangerute anamal of all to killing something gives me the moat thrilling experence it is even better than getting your rocks off with a girl the best part of it ia thae when I die I will be reborn in paradice snd all the I have killed will become my slaves I will not give you my name because you will try to sloi down or stop my collecting of slaves for my afterlife ebeo riet emeth hpiti

我喜欢杀戮，因为这很有趣，比在森林里杀野生动物还有趣。因为人类是最危险的动物。比起和女孩在性爱中获得高潮，屠杀能给我最兴奋刺激的体验。最棒的是，ia thae，我死后会进入天堂。我杀死的那些人会成为我的奴隶。我不会告诉你们我的名字，因为你们会试图放慢或停止我为来生收集奴隶的脚步。ebeo riet emeth hpiti。

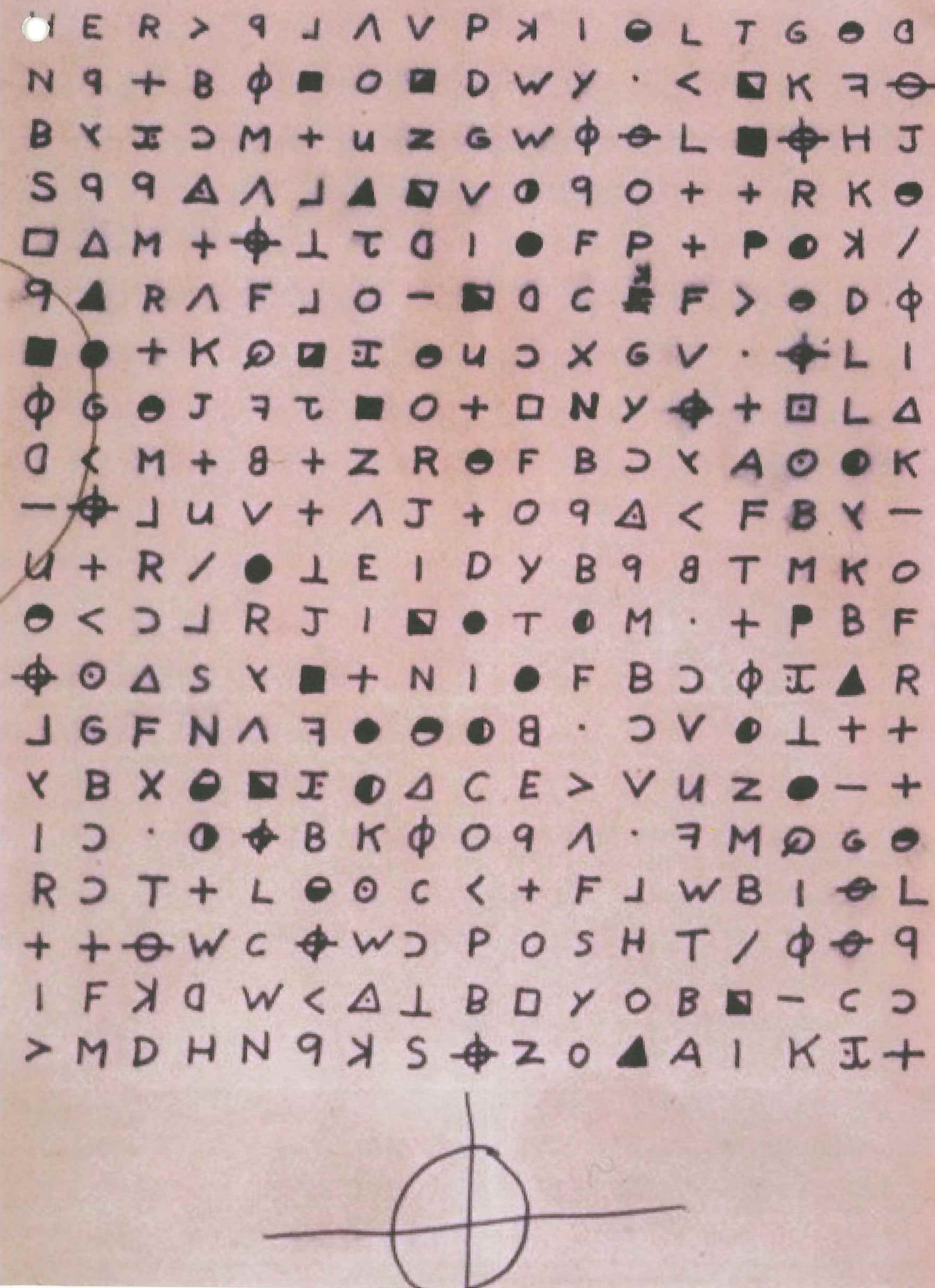
”

虽然有十几个符号并没有被破译出来，但这一结果很快被 FBI 确认。

很显然，Z408 被破译之后，Z340「升级换代」了。这一次的「升级」一度让破译陷入瓶颈。人们猜测，Z340 不仅使用了代换，还使用了移位。最难破解的恰恰是移位这一部分。

谁也不会想到，这一次的破译，竟用了 51 年。

Z340 直到 2020 年 12 月份才被破译成功。而在没来得及更新的出版物里，对 Z340 的描述仍旧是「未被破解」。



破译 Z340 的团队共有三个人，分别是澳大利亚的数学家 Sam Blake，比利时程序员 Jarl Van Eycke 及美国软件开发人员 David Oranchak。

针对这条只有短短 340 个符号的密文，Sam 总结出的所有移位可能共计 650 000 种。接着 Sam 和 David 分别将这些可能投入了 Jarl Van Eycke 开发的破译程序 AZ Decrypt 和 Brax Cisco et. al. 开发的 Zk Decrypto 中。让这些程序开始运行所有可能的移位方式。

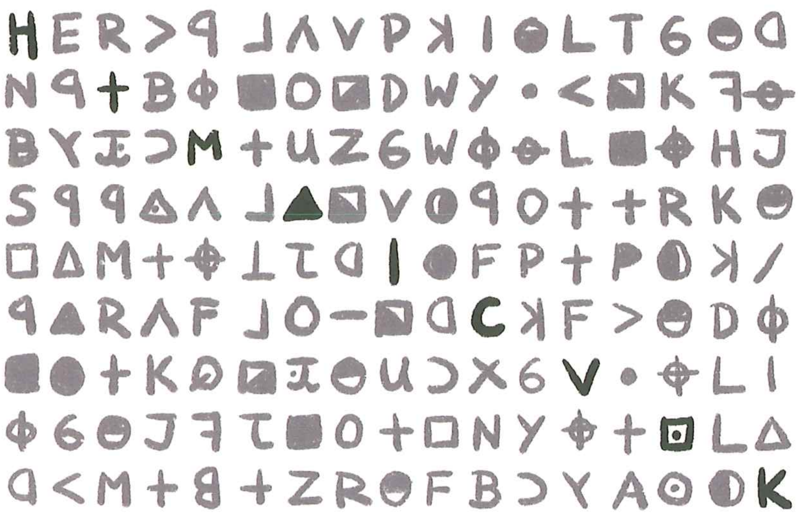
经过了数个月的运行，2020 年 12 月 3 日星期四，AZ Decrypt 上出现了一堆看似杂乱无章的英文字母，却吸引了 David 的注意。

> EHOPE YOU ARE HE SING IST TORRA ENN
> TRYING TO CATCH METH AFTAINT MT ON THE
> TS SHOT WHICH BRINGS UP ALS IN TABS IT ME
> NAME OF AR HEED OR THE GAS CHAMBER BECA
> ATE IT WILD VENT ME ROLER A DICE AI I THE VS
> SHEN BECAUSE TOO WHA SEEN TIGHT DESERTS
> WORS ROS ME THERE EVERYONEED HE HAS
> NOTHING THEN THEY HE ACH PARADICT IS
> THEY ALREAREAND NORDER THER AMEO EARRE
> AND BECAUITE IS YOT TV HAT MR NEWE ITLE
> NEVER IND BAEYN NEIA AT A HOE CDR PET

在这些随机的英文字母中，出现了几条有明确意义的短语，hope you are（希望你）、trying to catch me（试图抓到我）、or the gas chamber（或是毒气室）。其中的 gas chamber（毒气室） 是一条关键的线索，因为在报社收到 Z340 之前，有人冒充十二宫在电视节目上同一位律师对话，提到了这个词。

AZ Decrypt 是怎么得到这些短语的？为了一探究竟，David 看了 Sam 在位置上的猜测。他发现 Sam 将 Z340 分成了三个部分，第一部分和第二部分均为 9 行，第三部分为剩下的两行。

第一部分为对角线阅读，从左上角的第一个字母开始，对角线一直到右下方。





怀着激动的心情，David 将重新排列组合的顺序继续放进了 AZ Decrypt，但奇怪的事情发生了，不仅程序没有得到有意义的明文，就连之前的那些短语也再没出现过。也许是因为字符太多，在短时间内程序还没有遍历到那些短语的对应关系上。于是 David 将有意义的明文字母和对应的符号锁定住，让程序只遍历那些还没有被破解的符号。

这一次，AZ Decrypt 呈现的结果让 David 兴奋地大叫了起来。

> I HOPE YOU ARE HAVING LOTS OFF AN IN TRYING TO CATCH ME THAT WASNT ME ON THE TV SHOW WHICH BRINGO UP A POINT ABOUT ME I AM NOT AFRAID OF THE GAS CHAMBER BECAASE IT WILL SEND ME TO PAY ALL CE ALL THE

请注意这句话，that wasn't me on the TV show，恰好对应了之前提到的电视节目。接着 David 按照同样的规律将对应关系以及移位顺序作用到了第二部分。这一次得到的呈现结果如下：

> SOOHENBECAUSEEOOWHAVEENSUGHSLAVERTOW ORVFOVMEWHEREESERYONEELHEHASNOTHING WHENTHEYREACHPAYALICESOTREYALREAFAAIDI OFLETTHIFAMNOEAFREAIDBNCAUISEIVYOWTSHA TMRNEW

没什么实际意义，但好歹有一些可辨认的单词。

暂时没什么思路。于是 David 先跳过了第二部分，将注意力放在了第三部分的那两行上。将同样的对应关系，也就是密钥应用在第三部分后，David 有了惊喜的发现。

> EFILWILLEBNAEASYENONI
> ECIDARAPDEATH

I、will、easy 几个有明确意义的单词浮现在眼前。除此之外，还有 ECIDARAP 这个词。看似没什么意义，但如果反过来呢？PARADICE。正好符合第一封密文 Z408 中十二宫对「paradise」一词的书写习惯。如果按照这样的规律，将剩下的单词反过来。

> EFIL WILL EB NA EASY
> ENO NI ECIDARAP DEATH
>
> LIFE WILL BE AN EASY
> ONE IN PARADICE DEATH

这样第三部分就破译成功了。

好了，现在只剩下第二部分了。按照和第一部分同样的对角线阅读顺序，David 发现十二宫在书写密码的时候存在一些拼写错误。比如 oow 有可能是 now，ensugh 可能是 enough，fov 应该是 for。

S A A S O H O S H A C L I F E I S

S H O U V L R E N N E C E R O A A

M I E A O S E A V R E O N H S E F

A D N D I T H E E V F E E T T P O

A T F E O B V M E E N E O E L H H

I E R H R A T E N Y R N O S R V S

E E N Y A E A T A C O N B O U T M

O E R H G R D Y I H F A W E E W G

H H W W Y A W E I A D I R U T W C

一个有意思的场景出现了。那些拼写错误的字母 OSRV 全都在同一行。而修正之后的字母应该是 NOSR。
你有没有发现什么？
如果把目光往前移，你会发现 OSRV 之前的字母，恰好是 N。
David 将这个现象告诉了 Jarl Van Eycke，他发现只要把这一行的最后一个字母 H 往前挪，挪到第四个字母的位置，后面的所有字母依次往后移动一位，这个错误就被修正了。

S A A S O H O S H A C L I F E I S
S H O U V L R E N N E C E R O A A
M I E A O S E A V R E O N H S E F
A D N D I T H E E V F E E T T P O
A T F E O B V M E E N E O E L H H
I E R H R A T E N Y R N O S R V S
E E N Y A E A T A C O N B O U T M
O E R H G R D Y I H F A W E E W G
H H W W Y A W E I A D I R U T W C

同时密文右上方有一句有明确意义的话—— life is，那就说明按照对角线阅读的时候，应该跳过这句。

“

SOO HER BECAUSE E NOW HAVE ENOUGH SLAVES TO WORV FOR ME WHERE EVERYONE ELSE HAS NOTHING WHEN THEY REACH PARADICE SO THEY ARE AFRAID OF DEATH I AM NOT AFRAID BECAUSE I VNOW THAT MY NEW

”

这样的话，最后的信息便呈现出来了。
尽管前面 SOO HER 的意义还没弄明白，但大部分的密文已经被破译成功了。12 月 5 号周六上午，三个人将破译的成果发给了 FBI，很快得到了 FBI 的回复以及认证。
对于那个没有被破解的 SOO HER，FBI 也给出了自己的解释，他们认为 SOO HER 有可能是 sooner。
这样一来，三部分被完整破译出来了。让我们合起来看一下。

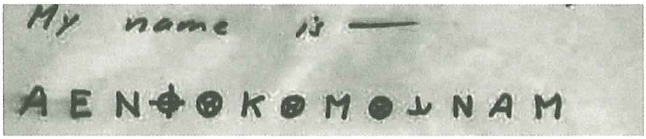
> I HOPE YOU ARE HAVING LOTS OF FUN IN
> TRYING TO CATCH ME THAT WASNT ME
> ON THE TV SHOW
> WHICH BRINGS UP A POINT ABOUT ME
> I AM NOT AFRAID OF THE GAS CHAMBER
> BECAUSE IT WILL SEND ME TO PARADICE
> ALL THE SOONER BEACUSE I NOW HAVE
> ENOUGH SLAVES TO WORK FOR ME
> WHERE EVERYONE ELSE HAS NOTHING
> WHEN THEY REACH PARADICE SO THEY
> ARE AFRAID OF DEATH
> I AM NOT AFRAID BECAUSE I KNOW THAT
> MY NEW LIFE IS LIFE WILL BE AN EASY
> ONE IN PARADICE DEATH
>
> 我希望你们抓我的时候能享受其中
> 出现在电视节目上的那个人并不是我
> 上面提到了我的一点
> 我不害怕毒气室
> 因为它马上就会送我去天堂
> 因为我已经拥有足够的奴隶为我工作
> 当到达天堂的时候 他们一无所有
> 所以他们害怕死亡
> 我不害怕因为我知道我的新生活是
> 在天堂里 生活会很轻松

Z340 虽然被破译完成了，但并没有什么有用的信息。「新生活」指的是什么？十二宫的真实身份又是什么？

值得一提的是，十二宫在 1970 年 4 月 20 日发出了一条密文，凶手用密码写出了自己

的名字，但至今未被破解。

对于 Z340 的破译成功，三个人很大程度上将其归因于运气 。而含有他姓名的这条密文,也许只有他本人才能告诉我们真相。



古典密码

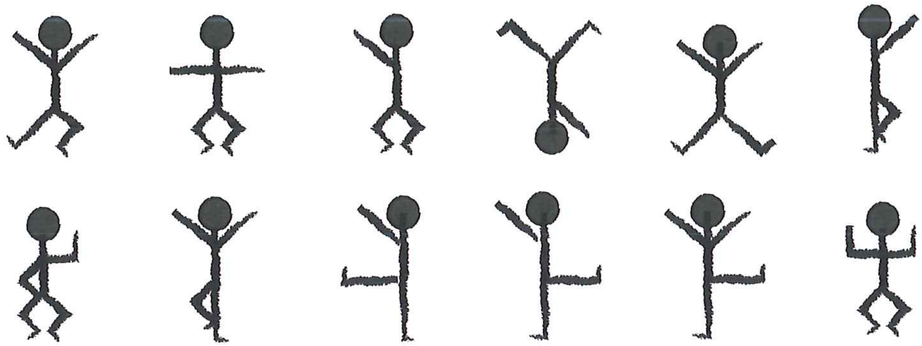
Classical Cipher

虽然古典密码在当代已经没有了用武之地，但它的「服役期」可以从公元前 5 世纪贯穿到上世纪 70 年代，被广泛应用在军事、政治等领域，与普通民众的生活较远。而我们现在使用的密码算法，也才发展了 50 年而已。

按照古典密码的加密类型，可以将它分为两类——替换密码和移位密码。替换指的是将明文字母替换成别的字母，顺序不变；移位密码顾名思义，指的是将明文字母改变位置的密码算法。当然也可以将替换和移位混合起来，形成一种更安全的密码。无论哪一类， 都只需要手、笔和纸就能完成。

还有一种不算是古典密码的隐写术，它并没有对信息进行加密，仅仅是将信息隐藏起来了而已。隐藏信息的方式多种多样，它和替换密码、移位密码一起，在历史的长河中扮演了非常重要的角色。

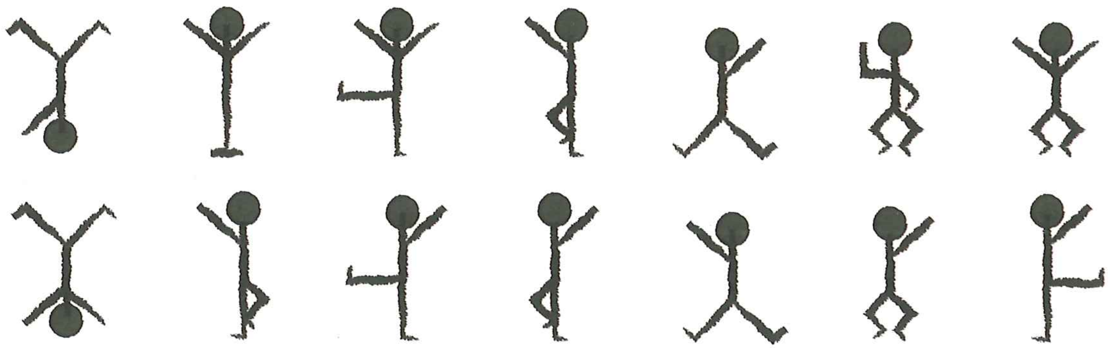
单表替换密码 Simple Substitution Cipher



单表替换密码应该是名气很大的古典密码了，在被破解前，它已经被使用了几百年。无论是《福尔摩斯探案集》里的《跳舞小人案》，还是爱·伦坡的《金甲虫》，都细致地描述过它。如果你是侦探小说的拥趸，那么对它一定不陌生。

单表替换密码的加密方式虽然很简单，但其密文表形式却多种多样，既可以是字母，也可以是奇形怪状的图形，还可以是符号，只要你愿意，怎么开心怎么来。这种多变的密文形式也给简单替换密码蒙上了一层好玩又神秘的气息。

但基本上无论什么形式的单表替换，都难逃被频率分析法破解的命运。



想要了解这个密码的更多细节，欢迎收看
正片**《黄金宝藏》**，有惊喜！

维吉尼亚密码 Vigenere Cipher

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

维吉尼亚密码是一种多表替换密码。它可以将一个相同的字母加密成多个不同的密文字母，弥补了单表替换密码在统计规律上的弱点，曾一度被称为「不可破解的密码」，被广泛应用于战争中。

但其诞生的 3 个世纪之后，陆陆续续产生了破译方法，比如卡西斯基实验和弗里德曼实验。

维吉尼亚密码是如何实现加密的？又是如何被破译的？

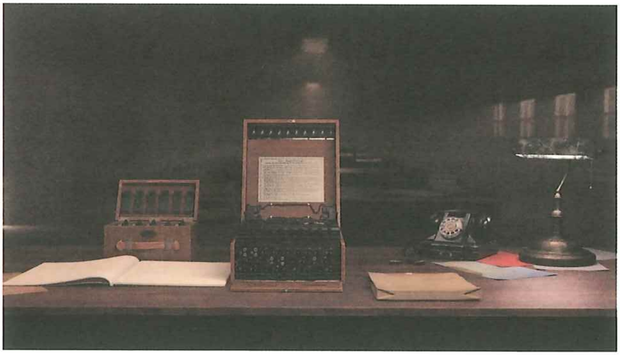
更多内容欢迎收看正片**《不可破解的密码》**。

恩尼格玛密码 Enigma Machine

单表替换密码、维吉尼亚密码和恩尼格玛密码这三者的发展历程非常直观地展现了密码算法是如何在「攻与防」的较量下不断迭代的。维吉尼亚密码可以抵御频率分析，恩尼格玛密码则加长了密钥的循环周期，弥补了维吉尼亚密码的缺陷。
正因如此，恩尼格玛密码的破译难度也比单表替换和维吉尼亚密码高得多。后两者均可用唯密文攻击破译，恩尼格玛密码却只能使用唯明文攻击。
恩尼格玛密码艰难的破译过程，即便是在波澜壮阔、战火纷飞的二战史上，也是值得大书特书的一抹光辉。

想要了解更多细节，请收看正片**《一个时代的终结》**。






Klappe
ENIGMA

波利比奥斯密码 Polybius Square Cipher

波利比奥斯密码由古希腊人发明，最早用于传递借助火焰信号。它由 5×5 规格的表格组成，行与列均由数字表示，25 个格子里都是字母。

	1	2	3	4	5
1	p	h	q	g	m
2	e	a	y	l	n
3	o	f	d	x	k
4	r	c	v	s	z
5	w	b	u	t	i/j

传递信息时，传送方左手举起的火把数量代表行数，右手举起的火把数量代表列数，行与列交叉的地方都是对应的字母。

波利比奥斯密码跟简单替换密码本质上是一样的，唯一不同的就是波利比奥斯密码会把 1 个明文字母替换为 2 个密文数字。比如加密「attack」，它的密文就是「22 54 54 22 42 35」。

凯撒密码 Caesar Cipher

凯撒密码的名字源于它的发明者——凯撒大帝。

凯撒密码加密的方式很简单，将整个字母表向左移动 3 位形成密文表：

A	B	C	D	E	F	G	H	I	J	K	L	M
E	F	G	H	I	J	K	L	M	N	O	P	Q

解密时只要反方向偏移即可。

这个过程也可以用取模运算去描述。首先将 26 个字母替换为 0~25 的数字：

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
R	S	T	U	V	W	X	Y	Z	A	B	C	D

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

然后运用加密公式 C=(P+K)mod 26 加密，其中 C 为密文，P 为明文，K 是密钥，为字母表的偏移量。

仿射密码 Affine Cipher

仿射密码和凯撒密码的加密公式有些类似，不过在公式中引入了一个乘数，即 C=(aP+b)mod 26，$1 \leqslant a \leqslant 26$，$1 \leqslant b \leqslant 26$。仿射密码的密钥为 a 和 b。解密公式为 P=a^-1(C-b)mod 26。

仿射密码涉及到了取模运算的乘法逆元问题，只有当 a、b 与 26 互质时，才可以正确地加解密。

这里涉及到了取模运算、乘法逆元等数学概念。

举个例子

假如要加密「attack」，替换成数字分别是 0、19、19、0、2、10。
我们取密码 a 为 3，b 为 7，则加密公式为 $c=(3p+7)\mod 26$。依次加密结果为：

明文字母	a	t	t	a	c	k
密文字母	H	M	M	H	N	L

3 在 26 中的逆元为 9，那么解密公式为

$p=9(c-7)\mod 26$

破译

仿射密码共有两种破译方式，它属于单表替换密码，也同样可以用频率分析破解掉。还有另一种方式。如果我们知道了两组明密文对儿，就可以通过一个方程组破译掉。比如说在密文里出现次数最多的字母就可以猜测它为 e，第二多的字母为 t，然后列一个方程组：

$$19=(4a+b) \bmod 26$$
$$13=(2a+b) \bmod 26$$

这个方程组可以进一步被推导为：

$$4a+b=26k+19$$
$$2a+b=26k+13$$

同时因为 a、b 要与 26 互质，且 K ≥ 0，且为正整数，那么所有的取值范围为 1、3、5、7、9、11、15、17、19、21、23、25，接下来给 k 取整数值，如果 a 与 b 属于这个范围，就说明破解正确。
在这个公式中当 k=0 时，方程有解。
即 a=3，b=7，破解完成。

我们在**《有限域》**这一节的视频里详细讲解了相关内容，欢迎收看！

希尔密码 Hill Cipher

希尔密码是 1929 年由数学家 Lester Hill 提出来的。在这个密码中，Hill 提出了用矩阵加密的方式。
希尔密码无法被唯密文攻击，只能通过已知明文攻击。

算法描述

将英文的 26 个字母用数字依次表示，需要注意的是，A 为 0，B 为 1，……

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

算法公式用矩阵表示，密钥矩阵根据加密的长度变化，假如一次加密 2 个长度，则密钥矩阵为 2×2；假如加密长度为 3，则密钥矩阵为 3×3：

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} \bmod 26$$

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

上面这个矩阵公式可以写为：

C=KP mod 26

这里的 K 便是密钥矩阵。值得注意的是，希尔密码并不是对称密码，也就是说，加密的密钥并不能用来解密。
为什么呢？让我们用公式推导一下。

$$C = KP \bmod 26$$
$$P = C \times K^{-1} \bmod 26$$

由此可知，要想解密，就需要求解 K 的逆矩阵。

加密

假设待加密明文为「attack」，替换为数字为 「0 19 19 0 2 10」。密钥为：

$$\begin{bmatrix} 7 & 3 \\ 7 & 4 \end{bmatrix}$$

依次的加密公式为：

$$\begin{bmatrix} 7 & 3 \\ 7 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 19 \end{bmatrix} \bmod 26 = \begin{bmatrix} 5 \\ 24 \end{bmatrix}$$

$$\begin{bmatrix} 7 & 3 \\ 7 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 19 \end{bmatrix} \bmod 26 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} 7 & 3 \\ 7 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 19 \end{bmatrix} \bmod 26 = \begin{bmatrix} 18 \\ 2 \end{bmatrix}$$

将这些数字换成字母，加密后的密文就是「FYDDSC」。

解密

解密需要密钥的逆矩阵，也称单位矩阵。因为逆矩阵的求解涉及到了与 26 互质的问题，这就意味着密钥矩阵并不能随随便便选择。
希尔密码需要模 26，也就是说 K*K^-1=1 mod 26，同理，下面两个矩阵相乘之后需要再模 26 才能得到 1 或者是 0，也就是说

我们要求解的是 26k 或者是 26k+1，其中 K ⩾ 0，且为正整数。

$$\begin{bmatrix} 7 & 3 \\ 7 & 4 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

逆矩阵是多少呢？你不妨求解一下吧！

一个小彩蛋

如果你看完 AES 密码的视频，就会发现希尔密码和 AES 的相似之处。
AES 和希尔密码一样，都是通过矩阵加密，再通过逆矩阵解密。不同的是 AES 的矩阵是固定的，而希尔密码的矩阵则随着密钥的变换而变换。

栅栏密码 Rail-fence Cipher

和上面描述的置换密码有些不同，栅栏密码是一种换位密码，也就是说，它并没有将明文字母换掉，而是通过某种规则打乱了明文字母的结构与顺序。

算法描述

我们假设栅栏密码的密钥是数字 n，那么把明文分成 n 行。比如「attack at down」，密钥为 3，即分成 3 行：

```
a...c...d...
.t.a.k.t.o.n
..t...a...w..
```

然后按照分割后的每一行重新组织密文，即为「acdtaktontaw」。

破译

栅栏密码的破解只能猜数字，但受加密方法的制约，栅栏密码的加密数字设定有两个限制，第一个是不能大于文本字母数，第二个是不要太大，否则加密分成很多行加密起来也很麻烦。

列移位密码 Columnar Transposition Cipher

列移位密码需要一个单词作为密钥。假设密钥单词长度为 n，就需要将明文转化成列为 n、行为 m（m 的数值根据明文的长度变化）的矩形。
接着按照密钥单词字母在字母表中的顺序，将列重新排列组合，再输出密文。

破译

列移位密码比较极端，如果密钥长度足够长，那么列移位密码就有很高的安全性（这个时候用电脑穷举攻击可能会更快一些），如果密钥长度足够短，人力就可以手动算出来。
但即便如此，它的破译过程也比想象的要麻烦一些，都需要依靠一定的蛮力去试。
第一种方式是猜密钥的长度，根据密钥的长度猜测所有可能的列的排列方式。但这

种方式有一定的限制，如果真正的密钥长度有 6 位，那么所有的排列组合多达 720 种，仅靠手动去算也是一个非常大的工作量。第二种叫做爬坡算法（Hill Climb），在密钥长度大于 10 的情况依赖电脑，进行启发式搜索进行破解。

Key Length	No. of permutations	Examples
2	2	AB,BA
3	6	ABC,BAC,CBA, ...
4	24	ABCD,ABDC,ACBD, ...
5	120	ABCDE,ABCED, ...
6	720	ABCDEF,ABDCFE, ...
7	5,040	ABCDEFG,ABDCGEF, ...
8	40,320	ABCDEFGH, ...
9	362,880	ABCDEFGHI, ...
10	3,628,800	ABCDEFGHIJ, ...
11	39,916,800	ABCDEFGHIJK, ...
12	479,001,600	ABCDEFGHIJKL, ...

藏头诗 Hidden Poems

提起藏头诗，很多人应该不陌生。最典型的藏头诗就是诗词每一句的第一个字连起来，形成一句有意义的话。其实藏头诗就是一种隐写术，将真正想要表达的意思隐藏在了诗词中。

在《水浒传》中，便有一首藏头诗：

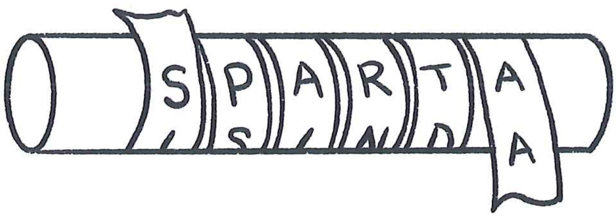
芦花丛中一扁舟，
俊杰俄从此地游。
义士若能知此理，
反躬逃难可无忧。

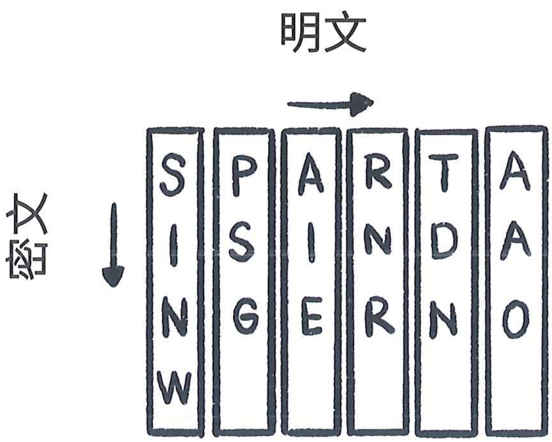
取头字「芦俊义反」。

密码棒 Scytale

密码棒可能是最早的隐写术方法。于公元前 404 年诞生于希腊。加密时需要用到一根棒子，一张写信息的纸条。将纸条缠在棒子上之后，在上面写下信息，然后将纸条取下来，这样的话原本的明文字母顺序就被打乱了。

要想还原明文，只要将这张纸条再次缠到同样粗细的棒子上就可以了。





培根密码 Baconian Cipher

培根密码诞生于 1605 年，它的发明者叫培根，对，就是你知道的那个培根。

加密

培根密码是一种用二进制数据加密混淆加隐写的密码，加密共分成两步：

1. 按照下面这样的格式替换每一个明文字母，可以将 a 和 b 换成任意符号，比如 0 和 1 等。

A = aaaaa	I/J = abaaa	R = baaaa
B = aaaab	K = abaab	S = baaab
C = aaaba	L = ababa	T = baaba
D = aaabb	M = ababb	U/V = baabb
E = aabaa	N = abbaa	W = babaa
F = aabab	O = abbab	X = babab
G = aabba	P = abbba	Y = babba
H = aabbb	Q = abbbb	Z = babbb

2. 替换完成后，将这个加密后的密文再换成有明确意义的另一条明文，但这条明文的加粗或大写字母要代表 b 或者 a，不加粗或者小写字母要代表另一个字母。

解密

一旦被知道采用了培根密码，就很容易被破解。只要按照大小写或者粗细体还原成二进制数据就可以破解了。

现代密码
Modern
Cryptography

20 世纪 70 年代，随着计算机的发展，古典密码逐渐被淘汰，现代密码学应运而生。同时，随着普通大众的日常生活越来越广地涉及到财产和信息安全之后，密码算法才真正地「飞入寻常百姓家」。

同古典密码一样，现代密码算法也分为两类——对称密码和非对称密码（公钥密码）。对称密码的设计准则来源于香侬提出的「混淆和扩散」概念，而非对称密码的设计则是来源于数学难题——素因数分解、离散对数求解。

对称密码又分为两类密码，一类是分组密码，一类是流密码。

分组密码只能按照固定长度分块加密明文，比如 DES 一次只能加密 64 bit，AES 一次只能加密 128 bit。

流密码是对「一次一密」的变相实现。一次一密使用的是异或算法，密钥是真正的随机数列，理论上不可破解，但现实中使用场景却非常受限，电脑也无法生成真正的随机数列。而流密码使用的是伪随机数列，密文长度和明文长度一样，「密钥种子」通过某种算法不断生成伪随机的密钥数列，与明文进行异或，输出密文。

对称密码和非对称密码各有优劣。对称密

码的运行速度更快，但与此同时，加密和解密用的都是同一把密钥，密钥如何安全有效地传送变成了一个问题；而非对称密码的加密和解密用的是不同的密钥，解决了密钥的传送问题，但巨大的运算量降低了它的运行速度。

因此现代网络安全采取的是两种密码算法的结合——非对称密码加密对称密码的密钥，对称密码加密信息本身。这样既保障了密钥的传送安全，又保障了信息加密的效率。

过于繁琐、计算量巨大的加解密步骤也决定了现代密码算法全部都只能在计算机上运行，单纯依靠纸和笔已经无法满足需求了。

但仅有这两种算法是不够的，对称密码和非对称密码只能保证窃取者不会知晓密文的真正含义，并不能保证密文不被篡改。

如何察觉出自己的密文是否被篡改了呢？

这就要用到哈希函数了。哈希函数就像文档的「指纹」，针对不同的文档可以计算出不同的散列值，哪怕只有一个字符的变化，散列值也会大不一样，这样就可以用来给每一个文档进行身份标识。

通过哈希函数，我们可以轻易地判别文档

是否被修改过。只要将面前文档的散列值与原本的散列值进行对比，如果一样，说明文档没有经过篡改，如果不一致，则说明这份文档不是我们要的那一份。

哈希函数严格来讲，并不是密码算法。但它却在现代生活中发挥着重要的作用，与现代密码算法一起构建了现代生活的安全网。

⚠ 现代密码部分读起来可能有一点点难，想要理解现代密码的类型和加密逻辑，欢迎收看正片！

DES

DES 的全称为 Data Encryption Standard（数据加密标准），是第一个由企业（IBM）和国家机构（NSA）一起开发的密码。在 20 世纪 70 年代投入使用后，一度为全世界所采纳和使用。

DES 使用了 Feistel 结构，每次加密 64 bit，共计 16 轮。Feistel 结构是对香侬「混淆和扩散」理论的具体实现，也是现代分组密码的基本设计准则。

加密时，Feistel 结构将明文等分成左右两部分，其中右半部分不做任何改动，直接进入下一轮，然后右半部分再经过轮函数输出，与左半部分异或后的结果进入下一轮。这样的操作会连续进行 N 轮，最终输出最后的密文。

Feistel 的这种结构保证了 DES 是可逆的，也就是说，加密和解密的步骤是一样的。

3-DES

1999 年，DES 被暴力破解，作为它的替补，3-DES 登场了。

3-DES 是 DES 的拓展，每次加密时，它会用三次 DES 处理明文，密钥长度也从 56 bit 扩展为 168 bit 。虽然 3-DES 的密钥变长了，可以抵挡暴力破解，但它依然每次只能处理 64 bit 的数据，效率很慢。

依然需要寻找更安全，更有效率的替代算法。

AES

AES 的全称为 Advanced Encryption Standard（高级加密标准），是替代 DES 和 3-DES 的密码算法。

和 DES 使用的 Feistel 结构不同，AES 使用的是 SPN 网络，每次可加密 128 bit 数据，密钥长度可为 128、192 或 256 bit。每一轮的数据不拆分，与密钥异或之后，直接进入 S 盒和 P 盒。重复 N 轮之后输出密文。

由于 AES 采用了严密的数学结构，它的加密算法与解密算法并不一样。

AES 于 2002 年成为新的加密标准，直至今日都是被广泛使用的分组加密算法。

RC4

RC4 密码算法由罗纳德·李维斯特在 1987 年提出，是流密码家族中非常典型的一个算法。

它的加密共分为两步。首先使用密码调度算法初始化与密钥相关的 S 盒，接着用伪随机生成算法生成密钥流。

RC4 在刚开始并没有被公开，而是 RSA 公司的机密。直到 1994 年，这个算法被匿名公开在了网络上，才被众人所知。

Salsa20

Salsa20 由丹尼尔·J·伯恩斯坦于 2005 年设计。

它的核心算法是一个由矩阵定义的函数 X，输入和输出均为 512 bit 。经过 20 轮迭代后，输出的结果与 X 进行模 2^32 加，生成密钥流。接着与明文异或，输出密文。

RSA

大数的素因数分解一直是一个世界性的数学难题。RSA 便是一个在此基础上设计出来的公钥密码。

它的加密公式为

$$c = m^e \bmod n$$

解密公式为

$$m = c^d \bmod n$$

其中公钥为 {e,n}，私钥为 {d,n}。e、d 和 n 的取值均是通过严密的数学公式算出来的，不能随便取值，而三者之中，只有 e 和 n 是公开的，而 n 就是保障 RSA 密码安全性的关键——它不能被找到可以分解的素因数，一旦被分解成功，整个密码都将被破译。

由于庞大的计算量，RSA 的运行速度相比对称密码会慢一些，这也是所有公钥密码的通病。在现实应用中，RSA 等公钥密码一般用来加密对称密码的密钥，而真正的明文依然由对称密码加密。

ElGamal

ElGamal 算法于 1985 年由塔希尔·盖莫尔提出，它的安全性基于在 mod N 的运算中难以求取离散对数。
它的加密公式为

$$a = G^k \bmod P$$
$$b = y^k M \bmod P$$

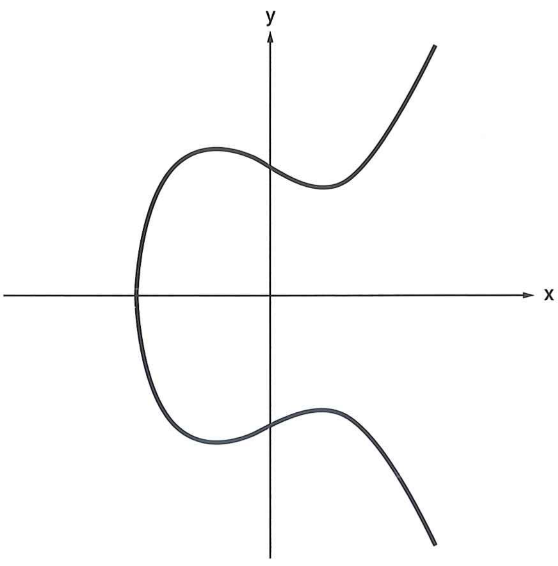
解密公式为

$$M(plain) = b \times (a^x)^{-1} \bmod P$$

其中 P 是随机选取的一个素数，G 为在 $\mathbb{Z}^*{}_p$ 下的生成元。x 为小于 P 的私钥，公钥 y 由私钥 x 计算得出，为 {P,G,y}。
由加解密公式也可看出，ElGamal 的密文会比明文长一倍，这也是 ElGamal 的一个缺点。

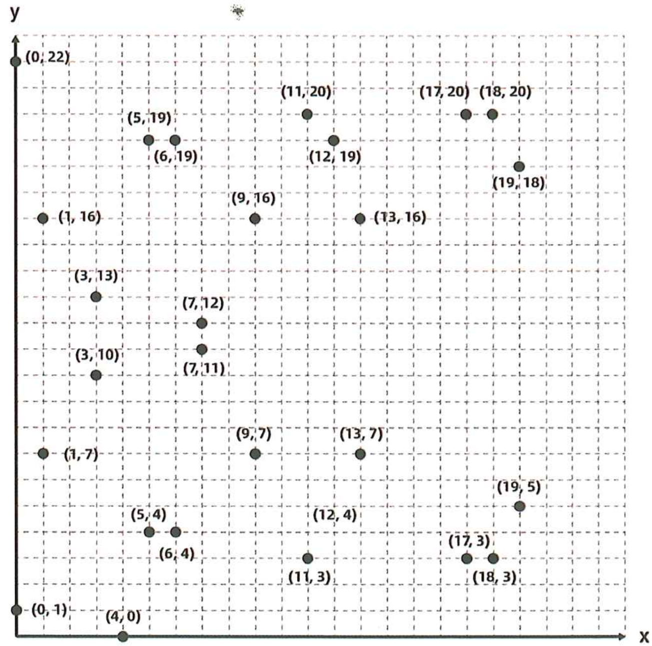
ECC

ECC（椭圆曲线密码）是继 RSA 之后被提出来的新的公钥密码算法。相比 RSA，椭圆曲线所需的密钥长度更短。
ECC 的安全性也是基于一种数学难题，那就是在有限域上的椭圆曲线上求取离散对数。
椭圆曲线并不是椭圆形状的，在实数域上，椭圆曲线长这个样子：



椭圆曲线示例 $E_1: y^2 = x^3 - 2x + 4$

而在有限域上，椭圆曲线不再是平滑的曲线，而是变成了一个个整数点：



椭圆曲线 E_2 在有限域 $\mathbb{F}_{23}$ 上的图像

简单来说，ECC 的安全性就在于在这个图像上，已知点 G 和点 xG 的情况下难以求取 x 的取值。

ECC 密码体制在很多领域都有广泛的应用，比如近些年大火的虚拟货币。

幕后故事
Behind-the-Scenes insights

幕后故事 1（制作人）

1

制作人的工作是什么？

刘叔叔：制作人类似一个项目的策划或导演，需要统筹和管理文稿、视觉、动效、三维和开发多个环节的进行中，为整体的项目负责。

2

你做这个项目的思路是如何开展的？

刘叔叔：我是中途接手这个项目的，作者同事已经做了较多的资料整理和初稿写作。密码学的内容并不简单，还是要很感谢作者很辛苦地啃了下来，保障了我们后续的工作。

但正因这些内容确实比较复杂，甚至可以说是无聊，所以怎么做一个整体的构思，让各个知识点之间产生联结——也就是所谓的作品感——很重要。于是我们中途经历了大幅度的文本重构。

最初这个项目可能更像是一门真正的密码学课程，但我还是希望它更有趣。刚好查资料，发现玛丽女王的故事在密码学里非常经典，而假如玛丽女王掌握了后面的密码，结局几乎就可以改写了，所以你可以理解成这是一条模糊的主线，为玛丽女王寻找更厉害的密码。当然它非常模糊，我们并没有强调。

真正的主题是密码朋克。原因并不复杂，这个概念确实很酷。密码朋克的精神让人向往，特别是在现如今这个毫无隐私可言的时代。密码学本身就没用，学了也不能去破解别人的密码，而且它那么复杂和乏味，普通人肯定没有耐心。所以我的想法是，明白这样一种概念，而不是真正去掌握它。谁需要去掌握它呢？搞网络安全的人。

这样一种故事性的设计，也算是基本操作的一次尝试。就看观众的反馈吧，如果大家都觉得课程应该更严肃，也许以后我们的制作思路也会转变。

3

这个项目跟传统的密码学课程有什么区别？

刘叔叔：最大的区别是我们想保证深度，但它又不能太无聊，我看了好几套系统的密码学课程，真的挺无聊的。所以从一开始我们就希望这项目能写实，不仅视觉沿故事的时间线进行了递进，增加场景感，在文本叙述风格和补充资料上，理论上都应该写实。比如频率分析，因为时间限制，我们只讲了一点点，但实际上我们应该补充更多关于频率分析的实战技巧。理想的状态是，观众把这个项目当成一个朋友在跟他讲一个有趣的事儿，而且听完后他还能真正上手试试，而不是在课堂上那样听一个理论。

其次游戏化的设计也是一个差别，比如《黄金宝藏》的寻宝交互，本质上跟密码学没啥关系了，但一个很重要的概念是，密码学并不只是密码算法本身，还关乎密码的使用，包括现代密码，现代密码已经那么安全了，所以如果一个人想攻破你的系统，他很少会直接对算法进行攻击，而会选择安全协议或其他的物理漏洞，甚至动用社会工程学。

4

哪一个章节做得比较痛苦?

刘叔叔：越往后越痛苦。先是 Enigma，文稿并不难搞，难的是视觉，但这玩意儿真的很绕，要在那么短时间内让观众完全看明白，难度是非常大的。现在的版本还有很多优化空间，因为绝大多数人想一遍看懂 Enigma 可能性还是比较小，但没办法，时间限制，我们也没法继续修改了。

像 DES 和 AES 都是很痛苦的。DES 的叙事思路，在作者的初稿基础上，我也仍花了一周多时间去构思。这种东西是这样的，讲明白是什么不难，难的是告诉别人，它为什么是这样的。AES 也很麻烦，因为 DES 把该讲的几乎都讲了，在这里如果只讲一个单纯的算法，会比较无聊。最后我们索性从第二次评审大会的背景出发，直接将 AES 做成了一个大交互，让观众自己去探索。纯属无奈之举。

5

自己最满意的是哪个章节?

刘叔叔：我个人挺喜欢《黄金宝藏》。这个视频很简单，几乎就是还原了爱伦坡的小说。频率破解的步骤看起来挺过瘾，最后一个交互几乎就是一个早期的网页解密游戏，就是好玩。不过 Enigma 那期也很酷，时间很长，十七分钟，动效师做了一个半月。

6

密码学项目还有后续更新章节吗?

刘叔叔：后续还有公钥密码的章节更新。密码学相关的知识点这个项目也只涉足了一小部分，如果观众的反馈不错，我们也会考虑更新网络安全和加密货币的内容。

幕后故事 2（作者）

1

为什么会想到要做密码学？

何呵呵：一开始想选题想到头大，后来有一天无意翻豆瓣，翻到了自己曾经给一本有关密码学的书标记过「想读」，当时觉得应该有搞头，于是查了好多资料，梳理下了大概的脉络发现密码学是可行的，于是和吴老师聊了聊，就决定做这个选题了。

2

密码学其实涉及到很多分支，现在的目录是怎么确定下来的？做了哪些取舍？

何呵呵：初版目录和大家现在看到的相差其实很大。一开始我主要是想要把重点放到现代密码学上，古典密码其实只有两篇，一篇讲简单替换，一篇讲 Enigma 。但是后来写完简单替换密码之后，发现并不能很顺畅地接到 Enigma 上，于是又补了维吉尼亚密码。

后面的现代密码学的内容也是根据密码算法的类型安排的，分别是对称密码和公钥密码。本来 RSA 密码之后应该是椭圆曲线密码，但是某一天突然灵光一闪，觉得玛丽女王的故事（也就是整个系列的第一集）完全可以引出哈希函数，碍于时间问题，只能把椭圆曲线去掉，改成哈希函数，算是一个首尾呼应？

还有就是《密码大厅》这个纯交互的章节。它本质上就是一个解谜游戏，有点像 Steam 上的 Cipher 这个游戏。当初想要搞这个的想法是，讲完古典密码直接讲现代密码算法好像有点无聊，因为古典密码其实大部分是可以被破译的，大家应该蛮喜欢解谜游戏的，于是我在里面设计了好多正片里没有讲到的各类古典密码的谜题，比如方格密码、移位密码之类的。

《密码大厅》其实还承载了另一个非常重要的功能。有很多有关密码学的重要的概念、发展之类的内容不太好做成视频，但是我们又觉得这个部分不能去掉，干脆就以收集卡片、墙上的背景字这种形式嵌进了密码大厅里，这样看起来也更丰满些。

最后三维老师和分镜老师呈现出来的效果非常棒。

3

写的时候主要从哪里获取信息？

何呵呵：古典密码其实比较好搞，因为内容很简单。

现代密码算法的话，主要就是大学教科书和网上的大学课件。我在二手书店买了一本密码学教材，买来的时候这本书已经是一副「惨遭蹂躏」的样子，后来被我们翻了无数次，感觉已经破上加破。

在此非常感谢两个网站，一个是 NIST 的官网，一个是 Stack Exchange 中的 Cryptography 这个板块。

写 AES 和 SHA-3 算法的时候，NIST 的官网真的帮了我不少忙。因为这两个算法都是 NIST 制定的标准，所以他们在官网上发布了非常非常多资料，包括算法详解、算法加密举例（具体到了每一步的 bit 变化）等等。

虽然有关密码学的资料很多，但其实有一些细节问题还是会被忽略掉。这个时候就要祭出 Stack Exchange 这个神仙网站了。你会发现不止你一个人有这样的问题！在 Cryptography 这个板块，会有各种有关密码学的算法细节问题，底下的回答有非常大的帮助！

当初我在 SHA-3 的 padding 规则这里困惑了好久，因为怎么也算不出 NIST 官方文档给出的结果。就是在 Stack Exchange 这里找到了一点思路。

感恩！

幕后故事 3
（分镜＆交互）

1

两位负责了哪些部分的分镜和交互？

Jialian：主要负责了玛丽女王到 Enigma 的分镜和交互。

Alaska：主要负责了 DES 之后的分镜。期间因为 Enigma 的交互有调整，我就也试着做了一下交互，和分镜相比是很不一样的感受。

2

是哪里不一样呢？

Alaska：两者考虑的内容不一样。分镜需要考虑文本逻辑和画面，而交互除了要考虑这些，还要考虑用户操作。这个交互的目标是什么？首先要在流程上满足这个目标，之后再考虑画面需要哪些功能按钮，按钮点击之前和点击之后的效果区分等等。

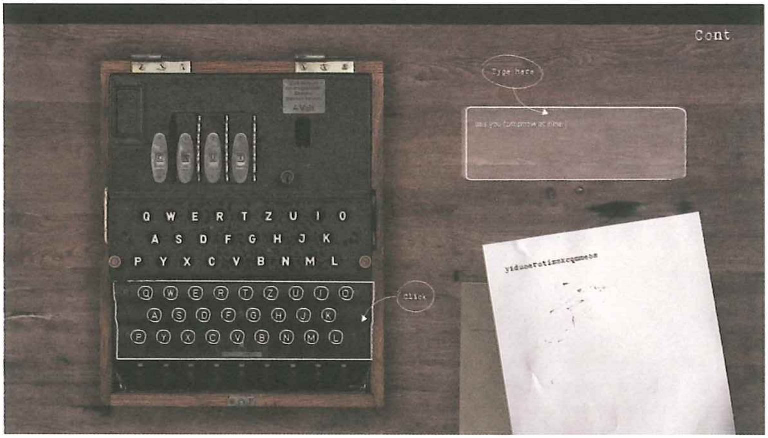
3

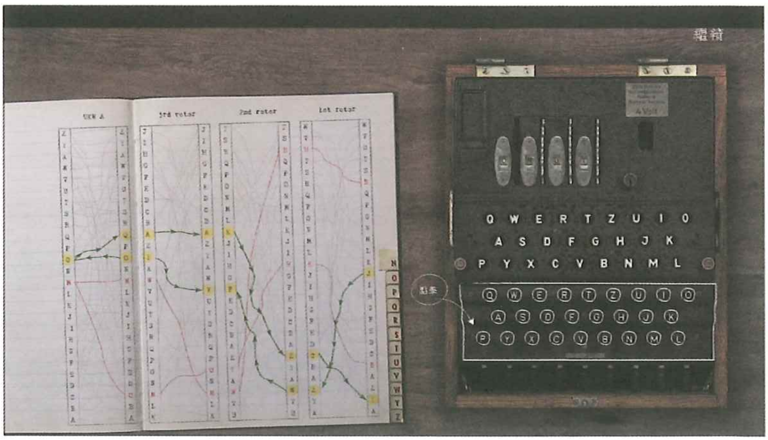
哪一部分的分镜 / 交互是印象最深刻的？

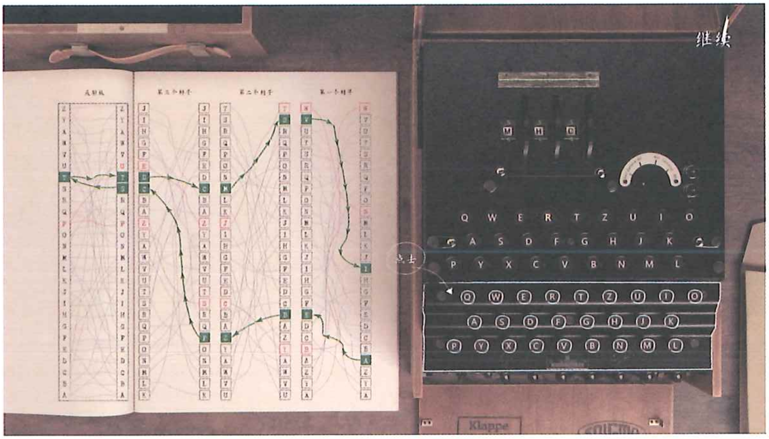
Jialian： Enigma 的分镜是最印象深刻的，因为我入职第一集做

拼贴场景 ▶

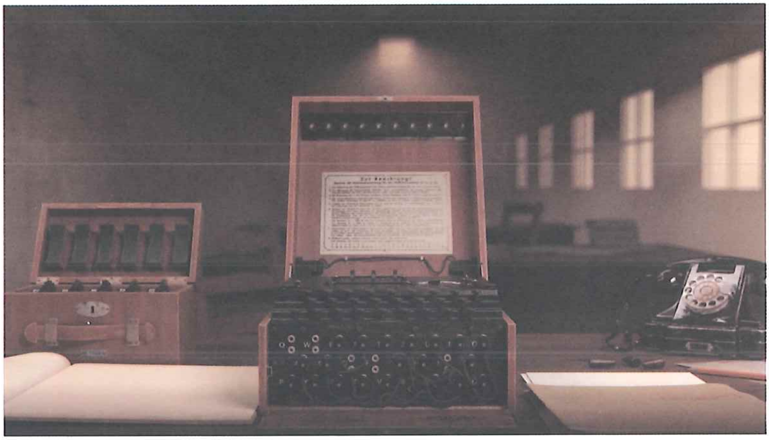

ESPAÑA
Click
Underwood Noiseless

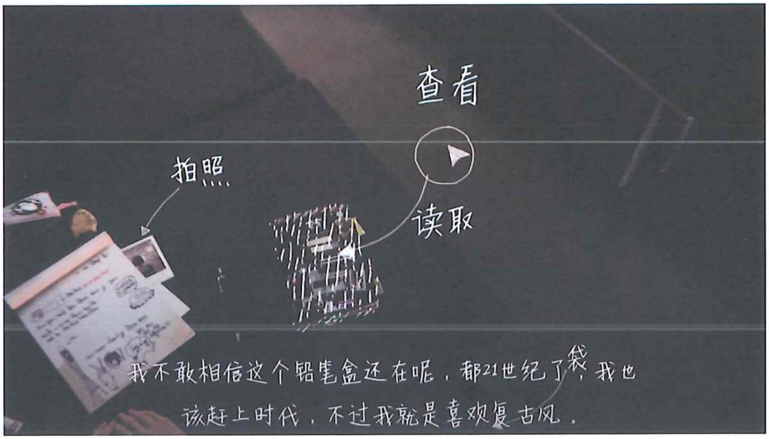

Cont
Type here
Q W E R T Z U I O
A S D F G H J K
P Y X C V B N M L
Click


3rd rotor
2nd rotor
1st rotor
Q W E R T Z U I O
A S D F G H J K
P Y X C V B N M L



◀ 建模后场景





◀ 《奇异人生》
游戏交互界面

的就是 Enigma，对于没有工作经验的人来说难度很大，也是我投入的精力也最多的一集。光是确定风格就好像就用了三周，也是在这几周里了解了这个公司的能力，熟悉了工作流程。后面就大胆地用拼贴拼出了一个三维场景，被三维老师完美超越得建了出来，就是现在看到的样子。

4

因为不同的密码算法涉及到不同的年代，整个片子的时间跨度非常大。在考虑不同章节的时候，视觉风格有什么参考或者灵感吗？

Jialian：因为是当做游戏来做的，所以各种游戏参考的比较多。比如《奇异人生》，我很喜欢它显示可交互物体时高亮物体的样子。此外有时也会根据密码出现的年代和稿子文笔给人的感觉来定风格。

Alaska：目前是根据年代或者电影来寻找相关的元素。比如 DES 是上个世纪七十年代出现的，所以就参考了七八十年代的电脑操作页面。

5

自己觉得最满意的部分是哪个？

Jialian：没有，设计这种东西就是永远可以更好。

Alaska：还没有做到觉得最满意的部分，做到后面也许会有。

6

分镜和交互做的最痛苦的是哪个?

Jialian：DES 的交互。

Alaska：DES 的分镜吧，当时相当于做的第一个分镜，从风格的确定上面就花了很长时间，终于确定了风格后，做到后面发现平面和黑白色的像素风格做起来限制很大, 关键是感觉也不够好看，就比较崩溃。

6

不管是分镜也好，还是交互也好，其实都是第一次做，在这个过程里有什么感想吗?

Jialian：累了求开。

Alaska：最大的感想就是如果一直对着自己的分镜久了就会失去判断能力，包括判断信息传递是否清晰准确，以及画面是否美观和谐，这种时候就需要别人来提提意见什么的，总能得到一些不一样的观点，也有助于自己换个思路。另外就是开始的时候看到自己的分镜经过动效老师的制作变成视频的感觉很神奇。

幕后故事 4

（动效）

1

可以说一下自己的分工吗？分别负责了哪些部分？

滨哥：我做的是《玛丽女王之死》《黄金宝藏》《不可破解的密码》《一个时代的终结》《分组！分组！》，还有后续的一些章节。

宋：负责了《密码标准的阴谋》这一集的动效部分。

梨花：番外有限域那集，部分文稿分镜的修改还有负责最后的成片。

2

有哪些动效做起来是很困难的？

滨哥：前面几个还好说，Enigma 的片长达到了惊人的 18 分钟！这种长篇的视频我犯了一个低级错误，起初剪辑音频的时候没控制好节奏，导致后面解析部分节奏过快，然后就是推翻重改。

其实动效还好，选音乐是真的要命，每次找音乐就是大半天，耳朵都快炸了，就这些吧。

宋：在后半部分展示 DES 原理中的核心，也就是轮函数部分是我遇到的比较困难的地方，在展示它结构的同时，还要把每个结构

的步骤细节完整的展示出来。

梨花：对于我那集，难点可能在文稿的部分，怎么提出有限域，找到个数是素数的规律，不是素数怎么解决，最后用多项式表达。牵涉到动效的话就是什么样的转场、视觉元素引导表现内容的逻辑，比如名词解释的部分相当于是一个额外的补充内容，应该跟主体描述区分开来，我用到的方法是反色表示，然后与之相关的例子也在黑色底上描述。

3

哪个动效是自己觉得非常炫酷的?

滨哥：比较喜欢的镜头是《玛丽女王之死》里面被水淹没那段，还有 Enigma 里面转子炸开的场景。在此感谢倩妮（三维）的模型场景提供。《黄金宝藏》里面有一个图书馆的假三维挺喜欢的。

宋：在讲到一次性密码本把一串长数据阵列式的分成 4 组 64 位的数据块那里觉得很炫酷。

梨花：这集动效都挺简单的，伽罗瓦域的表示形式觉得还挺合理的，倒也说不上炫酷。

4

密码学的工作，和你之前的有什么不一样吗？

滨哥：之前的项目可能听到标题就知道要做什么了，片子制作起来容易一些。但到了密码学就傻了，之前没有接触过类似的内容或游戏，初期制作都是糊里糊涂的，到后来发现密码学确实很有意思，逐渐熟悉起来就能找到其中的乐趣。以前做片子感觉每一集都是一部小短片，而密码学就像看复联一样，越看越快乐（不

会英文没法解密真难过），就这些。

宋：密码学的工作和之前的工作差距还是比较大的。整体的时长比较长，工作量比较大。最主要的，也是一开始遇到的最大困难，就是密码学的每个动效都是要讲逻辑的，需要直观地给观众展示出逻辑关系和知识原理，而不是像我之前做广告那样注重展示炫酷的动效和好看的画面。

梨花：跟节目组区别就是周期长，就更需要去扎实理解文稿的意思，前期也需要多参与讨论，做一些视觉风格上的尝试，毕竟是好几集的项目。

跟《一个人工智能的诞生》的区别可能是叙述风格导致的画面风格的不同。不过我做的这集啊，还是老本行，讲数学，不需要一个故事性的背景，所以我做起来的风格还是跟机器学习有点像。但因为文稿会比机器学习更长，把伽罗瓦域怎么推的过程基本上讲完整了，所以我这次动效节奏会做的比较慢，想尝试慢一点让大家能真的跟着思路去算。可能没那么爽，但如果一集体验完了能大概明白伽罗瓦域为什么会是这么算的。

幕后故事 5
（三维）

1

刚加入的时候，做的第一个三维是 Enigma，那个时候视觉是怎么确定下来的呢？

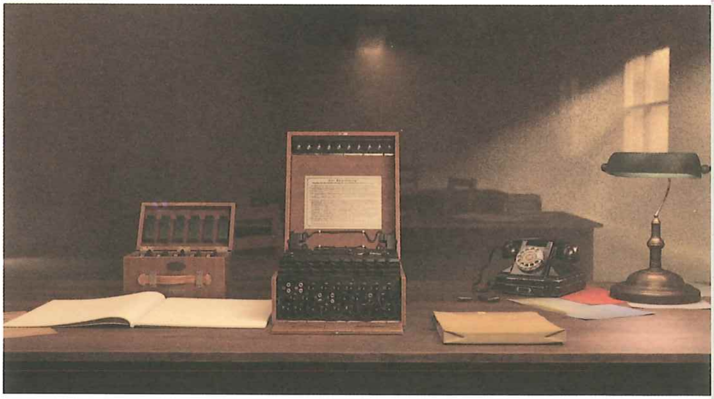
Achai：当时刚接到这个项目的时候，组长只是让做一个 Enigma 的键盘，完全没有做场景的计划，后来考虑到节目的趣味性加入了好多当时的历史背景，比如说二战。然后开始找各种历史参考，再加上分镜老师拼贴了一个场景，觉得很棒，但是 ... 一开始做出来的三维没有太强的空间感和景深，后来我尝试在背景上加长廊，再加上光效有点感觉了，然后就定了下来。

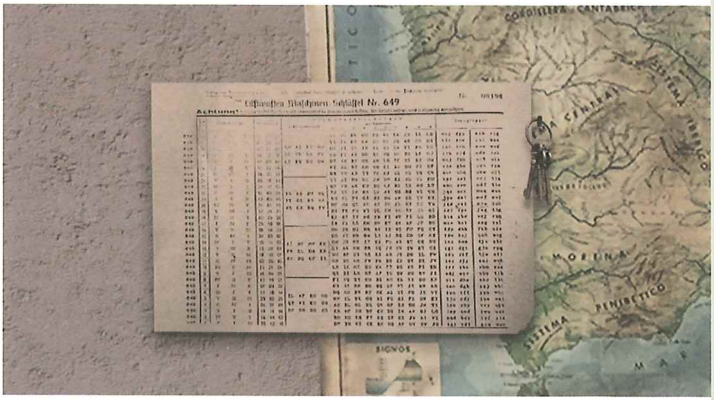
但 . 这只是它加密的场景，我们还需要加 Enigma 被破译的场景，因为历史原因，这两个场景是不一样的。所以要搭建破译的场景，历史上破译 Enigma 的主要场所就是布莱切利公园，我们参考了相关的历史图片，然后场景就搭建成了现在的样子。



Before ▶

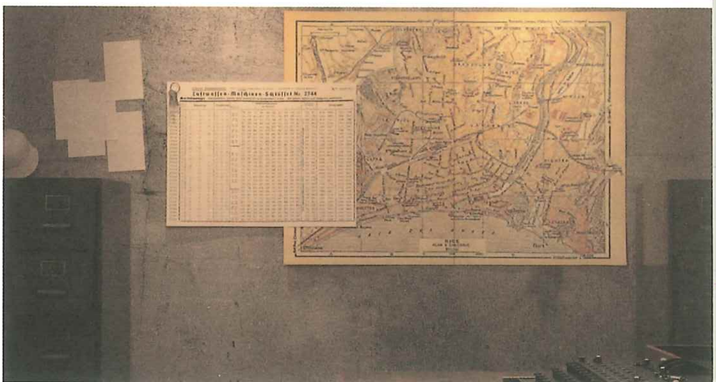
After ▶

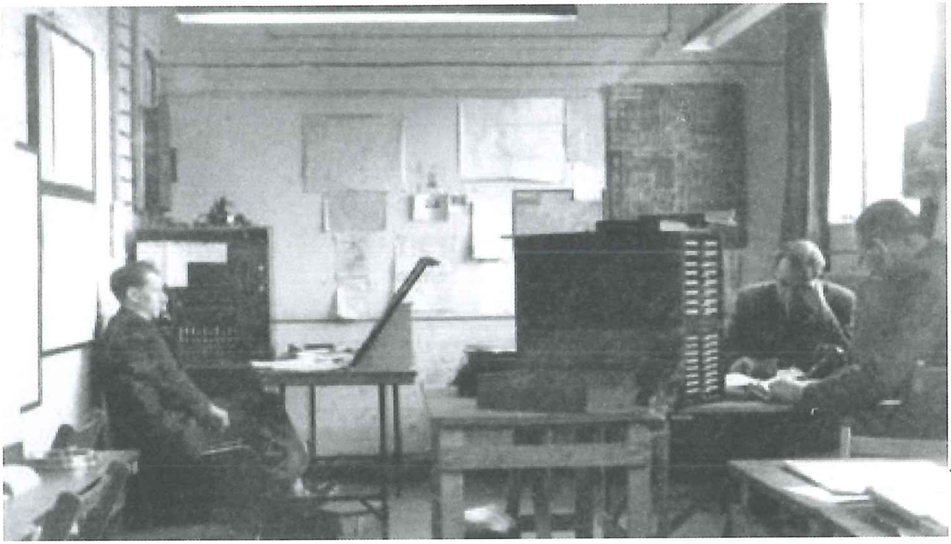




Before ▶

After ▶







▲ 布莱奇利公园
新闻 — 发现馆藏

▼ 布莱奇利公园
破解密码，1943 年





《一个时代的终结》分镜

2

有哪个三维是做得很开心的？

Achai：《密码大厅》和 DES，他们和以往的写实画风不一样，有更好的发挥空间和大胆的尝试，而且还可以探索未知的技术，然后再去攻略它，这大概让我产生快乐感吧！

3

未知的技术指的是什么呢？

Achai：就比如《密码大厅》这块，给开发老师物料之前，我还没有了解过烘焙贴图，以及要给开发老师什么样的三维格式，就去上网查，看别人怎么说，需要什么样的流程。后来另一位老师说需要 glTF 格式，我也导出了 glTF 格式，但是贴图导不出来，尝试了各种方法导出贴图（导出贴图还需要展 UV 等等），还是没有成功，后来我索性去官网上看，说是使用 PBR 方法来描述材料。我就在想是不是用默认的材质就可以了呢？然后就去测试了一个，果然成功了（激动）！

4

哪个三维做得很痛苦？

Achai：大概就是《黄金宝藏》吧 .... 树太多，面数也很多，卡到让人无奈。我从来没有做过室外的场景，还要考虑到地平线接缝的问题、天空的贴图等，好在我的师傅（我们的动效滨哥）帮我解决了这个问题，感谢我的师傅！

幕后故事 6（开发）

1

三位分别负责了哪部分的交互开发呢？

6dyallen：除了《不可破解的密码》，每集中都有部分交互点是我写的。

川凛：《玛丽女王之死》的第一个交互，《一个时代的终结》的最后一个交互，以及《密码标准的阴谋》的最后一个交互。

欢欢：除了 6dyallen 和川凛两位老师写的，排除留下的就是我写的 hhhh。《不可破解的密码》和部分《密码标准的阴谋》的第 2 个交互、DES 解密。

2

有哪些提交的交互设计方案和你们想象的是有出入的？在原有设计的基础上，你们有提出过哪些新的建议吗？

6dyallen：Enigma 这集最后找循环圈破译的那个交互和我想象的出入比较大。我当时建议保留前两个交互中左侧 Enigma 步进过程的那个图，但是最后没有被采纳。于是我选择了写这集的另外三个交互 hhh。

在《玛丽女王之死》这一集模拟计算机暴力破解的交互中，长按

按钮提高性能的功能是我提出来的。还有《黄金宝藏》第二个破译交互，本来当初设计时是两个交互，但我觉得合成一个比较好，中间用动效过渡，最终的效果还不错。还有后面要开发的 AES，听说要采纳我提出的滚动交互的方式，还挺期待的。

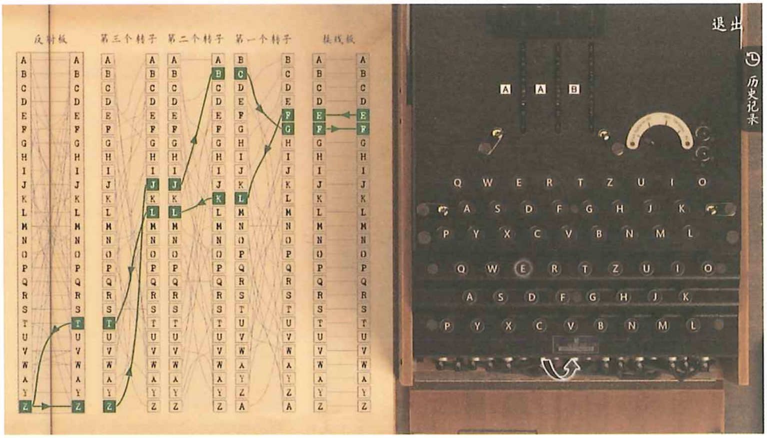
欢欢：我严重怀疑《密码标准的阴谋》是我职业生涯的一个大阴谋，一直踩坑。太过灵活的交互和规则都需要讨论设定，总觉得我就要做一个编译器了，真的不简单。感谢李老师帮我和视觉同事沟通，需求更加明确了很多。感谢钟老师（川凛）在交互实现的过程中给予了太多的帮助。感谢思密达。

川凛：密码学的大部分交互中，需要显示在同一个界面的信息都比较多，桌面版还好，移动端的体验就出现了一些问题，为了兼容移动端，我提出了如果交互检测到是移动端，交互就会从电脑版的布局动态过渡到移动端兼容的布局。但是个人觉得我们依然没有做到完美适配，之后也会继续探索。另一个比较头疼的交互就是《密码标准的阴谋》的最后一个交互，这个交互想展现 DES 密码的加密过程，因为加密步骤太多，又想让用户看到整个加密过程，也能看到每个加密环节的细节，所以，我们在交互的设计上纠结了很久。第一版想做成地图的样子，可以缩小看到宏观加密过程，也能拉近看到微观细节，但是第一版似乎少了很多乐趣，就像一个结构图，后来和徐老师、刘老师讨论之后，发现其实加密的过程，就像是工厂一步步加工货物的过程，最后这个交互就变成了观察工厂生产线。整个讨论过程虽然非常繁琐，但还是很有趣。

3

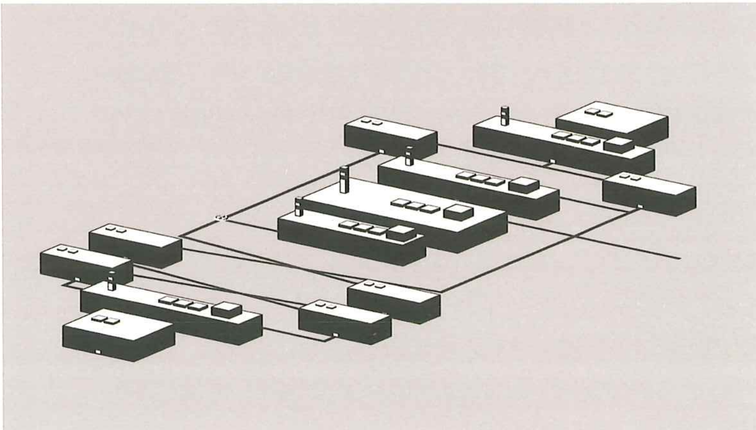
和前期所有环节一样，开发也需要对文本内容有一定的了解。迄今为止觉得哪个章节最难呢？对开发造成了什么影响？

6dyallen：DES 我听了两次，完全没听明白，和 Enigma 循环圈



左侧为轮子步进图 ▶

一轮加密 3D 版本 ▶



交互一样，我选择把它交给钟老师（川凛）写，感谢钟师。我负责的交互里倒也没有特别难理解的内容，不过有一个印象深刻的插曲， Enigma 的代码是我写的，钟师用的时候发现在输入相同的情况下，我写的程序和网上的那些 Enigma 模拟器的输出并不一样，但代码我觉得没有问题。最后查资料发现是 Enigma 异常的 Double Stepping 策略，在特殊情况下 Enigma 的中间转子就是会连续转两次。这样通过实践找到了密码机中隐藏的「秘密」，就非常有意思。

欢欢：密码学，越学越难的感觉。DES，也就是《密码标准的阴谋》我看了几遍，目前感觉应该是最难理解的。我只局限于粗浅的认识，没有时间和精力去深入了解。《密码标准的阴谋》交互 2 让我现在还在噩梦，真的有必要让用户自己画一个解码的过程嘛？至今仍然在思考会不会有更好的方式…我理想中的密码学更希望是一个密室逃脱的感觉。

川凛：我感觉最难的就是 DES 吧，主要是加密过程挺复杂的，还得了解每个环节都做了什么，并且需要将计算过程拆解。我也是在和设计、作者老师一起讨论的过程中慢慢理解的，我现在正在写这个交互，只能祝我好运。

基本操作
PLAY GROUND

项目说明手册 / 赠品 / 非卖品